

---

glossaries-ldf



DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Report for Practical Course

## **Control of Modular Robots**

Authors:	Authors
Supervisor:	Supervisor
Submission Date:	02.07.2015

# Abstract

In this course our team implements the dynamic model of Schunk's reconfigurable modular robot manipulator LWA 4P. We use Matlab and Simulink to implement the single modules. For that we first obtain the Equations of Motion using the Newton-Euler-formulation. We characterize the modules for an automatic modelling method by providing estimates for masses and inertia tensors using measurements and CAD data, so that the EoM can be automatically achieved. We then provide the preliminary Simulink model for the robotic arm. Secondly, we identify the dynamic model of the actuators. With Black-Box identification and with White-Box identification. In the next step, we implement a simulator of the robot by enhancing the simulink model including saturation effects and possible delays and validate the model using measurements. The fourth step is to design and tune the decentralized controllers while considering limitations due to the sampling time and test them on the robot. We finally test the controllers with a trajectory planner modelling the kinematic behaviour, which is developed by our second team.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Obtaining the Equations of Motion using the Newton-Euler formulation and creating a preliminary Simulink model of the robotic arm</b>	<b>2</b>
2.1 Obtaining the Equations of Motion using the Newton-Euler formulation	2
2.1.1 Theoretical background . . . . .	2
2.1.2 Implementing the equations in Matlab . . . . .	4
2.2 Implementing the Simulink Model . . . . .	5
<b>3 Deriving a dynamical model of the actuators and combining it with the Simulink Model of chapter 2.2</b>	<b>6</b>
3.1 Deriving a dynamical model of the actuators . . . . .	6
3.1.1 Theoretical background: DC motor . . . . .	6
3.1.2 Equations describing DC motor . . . . .	8
<b>4 Conclusion</b>	<b>10</b>
<b>List of Figures</b>	<b>11</b>
<b>List of Tables</b>	<b>12</b>
<b>References</b>	<b>13</b>

# 1 Introduction

As robots are getting more and more complicated, modular robots play an important role in robotics. Modular robots consist of independent cubes, which can connect themselves in different ways in order to fulfill a task. Each of the cubes has its own power supply and intelligence and can have specific tools for solving a problem under special conditions. After a computer has been fed a problem, it analyzes it and connects to the cubes in such a way that it can solve the problem. The cubes have the same size in modular robots in opposite to fractal robots, which can have cubes of different sizes. The cubes can be practically identical built, with similar software implementations, so that the input data and combinations of moduls determines how the task is executed.

## 2 Obtaining the Equations of Motion using the Newton-Euler formulation and creating a preliminary Simulink model of the robotic arm

Responsible: *Sophie Sepp and Benedict Feldotto*

### 2.1 Obtaining the Equations of Motion using the Newton-Euler formulation

In chapter two a preliminary Simulink model of the robotic arm is created using the Newton Euler Formulation of the Equations of Motion.

#### 2.1.1 Theoretical background

To derive the equations of motion of the manipulator we use the Newton Euler Method. The equations of motion are used to compute the dynamic model of the robot. While in the Euler-Lagrange Equations the manipulator is treated as a whole and the analysis is performed using a Lagrangian function, in the Newton-Euler formulation each link of the robot is treated in turn. Since each link is coupled to other links, the forces and torques of a link also appear in the equations describing its neighboring links. The whole process of deriving the terms of each link is called a forward-backward recursion. In the forward phase we obtain the rotational velocities and accelerations using the following equations, where  $i+1$  denotes the following link to the link  $i$ .

velocities and accelerations.jpg

$$\begin{aligned} {}^{i+1}\omega_{i+1} &= {}^iR \cdot {}^i\omega_i + \dot{\Theta}_{i+1} \cdot {}^{i+1}Z_{i+1} \\ {}^{i+1}\dot{\omega}_{i+1} &= {}^iR \cdot {}^i\dot{\omega}_i + {}^iR \cdot {}^i\omega_i \times \dot{\Theta}_{i+1} \cdot {}^{i+1}Z_{i+1} + \ddot{\Theta}_{i+1} \cdot {}^{i+1}Z_{i+1} \end{aligned}$$

Figure 2.1: Rotational velocities and accelerations.

accelerations of single joints.jpg

$${}^{i+1}\dot{v}_{i+1} = {}^i{}^{i+1}R(\dot{\omega}_i \times {}^iP_{i+1} + \omega_i \times (\omega_i \times {}^iP_{i+1}) + \dot{v}_i)$$

Figure 2.2: Linear accelerations of single joints.

For the linear accelerations of the single joints we use the formula:

To find the linear acceleration of the center of mass of each link we use:

acceleration for the center of mass of each link.jpg

$${}^i\dot{v}_{C_i} = \dot{\omega}_i \times {}^iP_{C_i} + \omega_i \times (\omega_i \times {}^iP_{C_i}) + \dot{v}_i$$

Figure 2.3: Linear acceleration for the center of mass of each link.

In the backwards phase we use the following equations to compute the forces and torques at the center of mass of each link and the forces and torques acting on each joint:

$$F = m\dot{v}_C$$

Force.jpg

Figure 2.4: Equation Force.

$$N = {}^C I \dot{\omega} + \omega \times {}^C I \omega$$

Torque 1.jpg

Figure 2.5: Equation Torque 1.  
and torques acting on each joint.jpg

$$\begin{aligned} {}^i f_i &= {}^{i+1}R \cdot {}^{i+1}f_{i+1} + {}^i F_i \\ {}^i n_i &= {}^i N_i + {}^{i+1}R \cdot {}^{i+1}n_{i+1} + {}^i P_{C_i} \times {}^i F_i + {}^i P_{i+1} \times {}^{i+1}R {}^{i+1}f_{i+1} \end{aligned}$$

Figure 2.6: Forces and torques acting on each joint.

### 2.1.2 Implementing the equations in Matlab

We first create a Simulink model of three links connected to the given initial constants and to the end effector. Each matlab function block holds a function to compute with the given input parameters the output parameters which become the input parameters of the next link. Thus all links are connected and compute the next values according to the input values they get from the previous link. This is done for the forward phase as well as for the backward phase. In the forward phase the linear and rotational velocities and accelerations of each link are computed. In the backwards phase the forces and torques that act on the centers of mass of each link are computed ( $F_i$ ,  $N_i$ ) and the forces and the torques that act on the single joints ( $f_i$ ,  $n_i$ ).

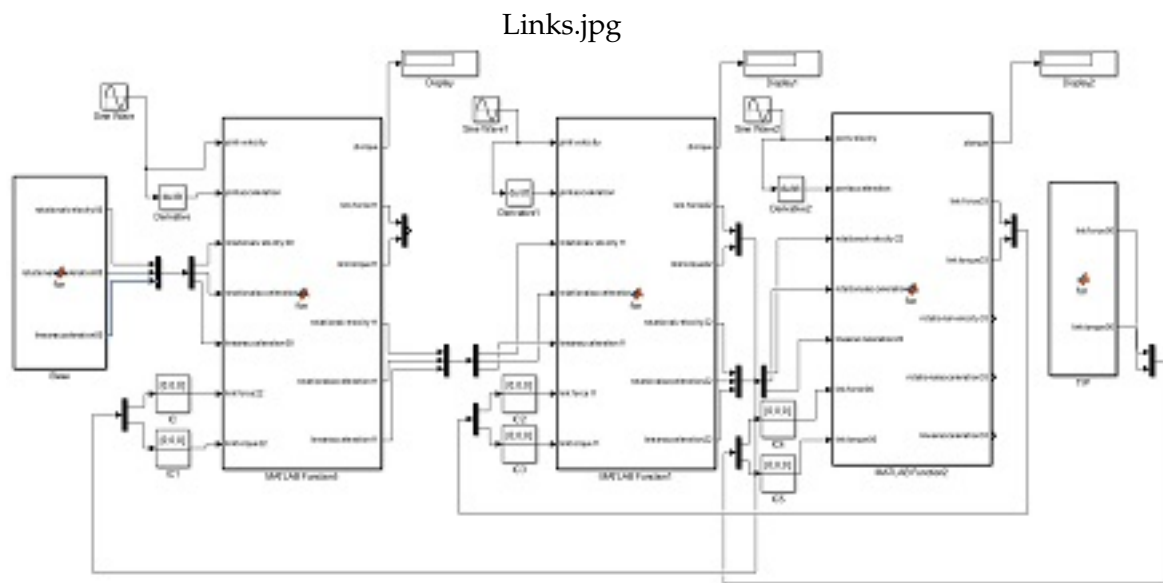


Figure 2.7: Three Links.



## **2.2 Implementing the Simulink Model**

## 3 Deriving a dynamical model of the actuators and combining it with the Simulink Model of chapter 2.2

Responsible: *Sophie Sepp*

In task 2 a dynamical model of the actuators is derived and the Simulink model of task 1.ii is enhanced with the actuator models. The robotic arm LWA 4P has six brushless DC motors which are modelled in Simulink and combined with the Simulink model of task 1. The modelling of the motor is hereby divided by the electrical part of the motor equations and the dynamical part of the motor equations.

### 3.1 Deriving a dynamical model of the actuators

#### 3.1.1 Theoretical background: DC motor

A DC-motor works on the principle that a current-carrying conductor in a magnetic field experiences a force, which can be expressed as the product of magnetic flux and the current in the conductor. It consists of a fixed stator, which provides a constant magnetic field and a armature part, which is a rotor that rotates inside the stator and is a simple coil. The armature is connected to a DC power source through a pair of commutator rings. The current that flows through the coil induces a electromagnetic force on it according to the Lorentz force, which causes the coil to rotate. Thus the commutator rings connect with the power source of opposite polarity. As a result, on the left side electricity flows away while on the right side, electricity flows towards. Thus the torque action is in the same direction throughout the motion, which is the reason why the coil continues rotating. A characteristical aspect of DC motors is the production of back emf. A rotating loop in a magnetic field produces an emf according to the principle of electromagnetic induction, which are in this case the armature loops. So an internal emf will be induced, that opposes the applied input voltage.

*3 Deriving a dynamical model of the actuators and combining it with the Simulink Model of chapter 2.2*

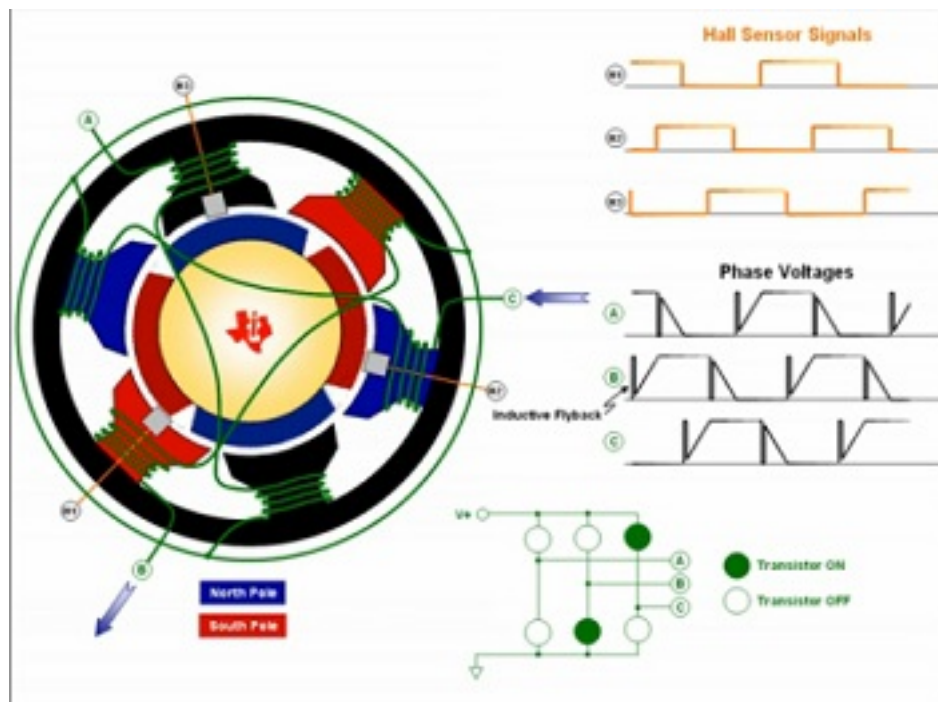


Figure 3.1: DC motor.

### 3.1.2 Equations describing DC motor

Electrical part of the motor equations:

diagram for armature controlled DC motor.jpg

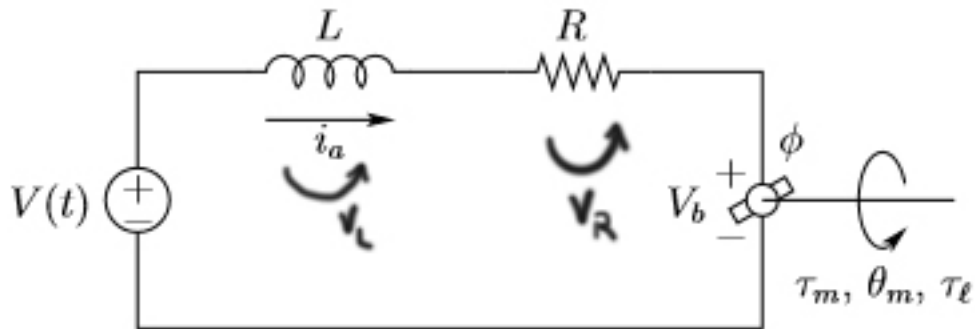


Fig. 7.3 Circuit diagram for armature controlled DC motor.

Figure 3.2: Circuit diagram for armature controlled DC motor.

$V(t)$ : Applied voltage  
 $L$ : Armature self inductance  
 $R$ : Armature resistance  
 $V_b$ : Back emf constant  
 $\theta_m$ : mechanical rotor angle  
 $\tau_m$ : generated torque  
 $\tau_l$ : load torque

The voltage  $V_L$  and  $V_R$  can be computed in the following way:

$$V_L = L \cdot \frac{di}{dt}$$

$$V_R = R \cdot i$$

### 3 Deriving a dynamical model of the actuators and combining it with the Simulink Model of chapter 2.2

The electrical part of the motor can be computed by the sum of  $V_l$ ,  $V_r$  and the back emf.

$$V_t(t) = L \cdot \frac{di}{dt} + R \cdot i(t) + V_{emf}$$

The back (induced) electromotive force  $V_{emf}$  is proportional to the angular velocity  $\omega(t)$  seen at the shaft.  $V_b$ , the emf constant, also depends on certain physical properties of the motor.

$$V_{emf}(t) = V_b \cdot \omega(t)$$

The mechanical part of the motor equations is derived using Newton's law:

$$J \cdot \frac{d\omega}{dt} = \sum_{i=1}^n \tau_i = -V_b \cdot \omega(t) + K_m \cdot i(t)$$

The inertial load  $J$  times the derivative of angular rate equals the sum of all the torques about the motor shaft and can be expressed by the sum of the induced current  $i$  times the torque constant  $K_m$  and the negative product of the back emf constant  $K_b$  and the angular velocity.

Moreover, the torque  $u$  seen at the shaft of the motor is proportional to the current  $i$  and can be expressed as the product of the current  $i$  and the torque constant  $K_m$ , which is related to physical properties of the motor, such as magnetic field strength.

$$u(t) = K_m \cdot i(t)$$

## 4 Conclusion

...

## List of Figures

2.1	Rotational velocities and accelerations. . . . .	2
2.2	Linear accelerations of single joints. . . . .	3
2.3	Linear acceleration for the center of mass of each link. . . . .	3
2.4	Equation Force. . . . .	3
2.5	Equation Torque 1. . . . .	3
2.6	Forces and torques acting on each joint. . . . .	3
2.7	Three Links. . . . .	4
3.1	DC motor. . . . .	7
3.2	Circuit diagram for armature controlled DC motor. . . . .	8

## List of Tables



# References

- [Lam94] L. Lamport. *LaTeX : A Documentation Preparation System User's Guide and Reference Manual*. Addison-Wesley Professional, 1994.